Student Name: Jonathan Riordan

Student ID: C13432152

Output of running the code

Plain text jonathanriordan

One Time Pad:

['s', 'a', 'l', 'c', 's', 'x', 'u', 'c', 'e', 'k', 'h', 'x', 'e', 'p', 'k', 'l', 'u', 't', 'h', 'f', 's', 'o', 'f', 'f', 'a', 'w']

[9, 14, 13, 0, 19, 7, 0, 13, 17, 8, 14, 17, 3, 0, 13]

[2, 14, 24, 2, 12, 5, 20, 15, 21, 18, 21, 15, 7, 15, 23]

Cipher text coycmfupvsvphpx

Decryption: jonathanriordan

The one time pad will change each time the program is run. It generates an array of letters from the alphabet. In order to decrypt. the person who wants to decrypt the message must have the one time pad alphabet from the encryption. the first array in the output is the one time pad, the second array is the plaintext letter position in the array. The third array is the cipher text letter position.

Code

#!/usr/bin/python

'''

Lab 6

Student Name: Jonathan Riordan

Student ID: C13432152

'''

from random import randint

keys = []

m = 11 \* 19

xi = 9

plainText = "Jonathan Riordan"

encrypted = []

alpha = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

def decrypt(cipherText, pad):

dec = []

plain = ""

decrypted = []

# get encrypted message positions

for i in cipherText:

if i in alpha:

positon = alpha.index(i)

dec.append(positon)

length = len(dec)

counter = 0

while counter < length:

messageNumber = int(dec[counter])

keyNumber = int(pad[counter])

total = messageNumber - keyNumber

if total < 0:

remainder = total

total = 25

total = total + remainder

decrypted.append(total)

counter += 1

for i in decrypted:

postion = int(i)

plain += alpha[postion]

return plain

def encryptMessage(messageIndex, keyIndex):

encrip = []

cipher = ""

length = len(messageIndex)

counter = 0

while counter < length:

messageNumber = int(messageIndex[counter])

keyNumber = int(keyIndex[counter])

total = messageNumber + keyNumber

if(total > 25):

temp = total - 25

total = temp

encrip.append(total)

counter += 1

print encrip

for i in encrip:

postion = int(i)

cipher += alpha[postion]

return cipher

def createPad():

pad = []

random = 0;

counter = 0

# Generate one time pad to be used.

while(counter <= 25):

random = (randint(0,25))

var = alpha[random]

pad.append(var)

counter += 1

print "One Time Pad:"

print pad

return pad

def removeCharacters(message):

message = ''.join(message.split())

return message

def encrypt(plainText, keys):

print "Plain text " + plainText

key = []

keyIndex = []

messageIndex = []

counter = 0

countKeys = 0

# Create pad

oneTimePad = createPad()

message = removeCharacters(plainText)

# Get plaintext character index

for i in message:

if i in alpha:

positon = alpha.index(i)

messageIndex.append(positon)

# Encrypt the plain text

print messageIndex

#Get one time pad index

for i in oneTimePad:

if i in alpha:

positon = alpha.index(i)

keyIndex.append(positon)

cipherText = encryptMessage(messageIndex, keyIndex)

return cipherText, keyIndex

cipherText, pad = encrypt(plainText.lower(), keys)

print "Cipher text " + cipherText

decryptedMessage = decrypt(cipherText, pad)

print "Decryption: " + decryptedMessage